AI Literacy Development for Software Engineering Education: Mapping AI from the SWEBOK to the SAIL Framework

David Parsons
The Mind Lab
academyEX
Auckland, New Zealand
https://orcid.org/0000-0002-9815-036X

***Abstract*—Artificial Intelligence (AI) is becoming a fundamental component of software engineering, both in terms of using AI applications in software engineering (AI for SE) and using software engineering to develop AI systems (SE for AI). The SWEBOK highlights relevant AI knowledge areas, but in practice, AI impacts all aspects of software engineering. As a result, software engineering education needs to take account not only of a set of knowledge areas but also a foundational set of AI literacies that inform students about the wider and deeper implications of AI systems. This article addresses this challenge by mapping knowledge areas from the SWEBOK to the Scaffolded AI Literacy (SAIL) Framework. Using some example mappings from knowledge areas to AI literacies, followed by illustrative examples of teaching scenarios, it explores how SE knowledge areas can be supported and enhanced by developing AI literacies.**

Keywords*—*Artificial Intelligence literacy, SWEBOK, SAIL framework, software engineering education, learning activities

I. Software Engineering Education and AI Literacy

With the increasing development of AI systems, and the widespread use of AI tools in the development of software, software engineering education must adapt to this significant change in the industry by integrating relevant coverage of AI into the curriculum [1]. Software engineering students need to develop an understanding not only of how to use AI tools within the software engineering process, but also how to apply these tools critically, appropriately, and with an awareness of both the opportunities and the risks of relying on AI technology. Further, it is increasingly likely that software engineering students will find themselves orienting towards future careers that demand a deep understanding of how AI systems are architected, developed, and evolved.

A. AI Automation in Software Engineering

As software engineers increasingly use AI tools to assist them in the software engineering process, some areas of activity may become fully automated while others will continue to require some human perception [2]. Areas less likely to see full automation are software requirements, design, and construction. In contrast, software testing and maintenance, software engineering management and process, models and methods of software engineering, and software quality, have all been suggested as candidates for full automation, alongside software configuration management, which is already a highly automated task.

DOI reference number: 10.18293/SEKE2025-011

These moves towards AI automation of the software development process have implications for the development skill sets required for software engineers across the industry.

Where software engineers work in the construction of systems that incorporate AI or machine learning, a wide range of challenges have been reported that are related to areas of development such as testing, software quality, data management, model development, project management, and requirements engineering, among many others [3]. Among the suggested software engineering practices that can help address these challenges are guidelines, lessons learned, and tools. In the category of guidelines, a combination of the SWEBOK and an AI literacy framework can provide a useful resource.

B. AI for SE and SE for AI

The two aspects of software engineering and AI (using AI and creating AI) are recognised in the Guide to the Software Engineering Body of Knowledge (SWEBOK) as AI applications in software engineering (AI for SE) where AI is used to help build software systems by taking over some development tasks across multiple development stages, and software engineering for AI systems (SE for AI), where developers learn the behaviours of AI systems and the role of training data [4].

In the context of SE for AI, the SWEBOK highlights the need for interdisciplinary collaborative teams of data scientists and software engineers, skills in evolving software that uses large and changing datasets, relevant software design patterns, and ethics and equity considerations, a broad skillset that goes beyond tools and technologies to embrace broader literacies. This is in a context where the methodologies, tools and practices of AI Software Engineering are still developing and much less mature than those of traditional software engineering [5].

C. AI Literacy to support Software Engineering Education

Given these significant changes to the role of the software engineer, students in the field will need to develop not just new knowledge but new AI literacies. These literacies include understanding AI concepts and developing technical skills with AI tools, as well as developing cognitive skills and AI digital citizenship, with a critical awareness of ethical considerations and the responsible use of AI technologies.

To support the goals of the SWEBOK in helping students develop their knowledge and skills in AI in software engineering, this article applies the Scaffolded AI Literacy (SAIL) framework [6] to provide a guiding structure for AI literacy in software engineering education. All the competencies referred to in the examples in this article are drawn from this framework.

SAIL is not the first AI literacy framework to be discussed in the context of technology education, but previous examples have been focused in specific knowledge areas, directed at school students, and/or lack detailed competencies that support scaffolded development [7]. The SAIL framework was developed from a Delphi Study of experts in 2024 and is designed to support all learners across multiple domains and levels of study. This division into multiple layers makes it relatively unusual among AI literacy frameworks, and its consideration of what comes beyond literacy, in the expert realm, males it uniquely relevant to discussing AI capabilities at the graduate level. Although the framework is designed to be generally applied across all learning domains, its focus at the higher levels on the knowledge, skills, and understanding that are needed to work in depth with AI technologies makes it an appropriate vehicle for supporting the examples explored in this article, which address scaffolded AI literacy development in software engineering education.

The remainder of this paper is structured as follows. First, it reviews the current coverage of artificial intelligence in the SWEBOK. It then briefly introduces the four levels of the SAIL framework, suggesting target competencies for software engineering students and graduates. This is followed by two examples of applying different levels of the framework to knowledge areas of the SWEBOK. In each case, there is some discussion of how this alignment with AI literacies might deepen and help contextualise the technical knowledge that students will be addressing from the SWEBOK. This is followed by two examples of how learning activities might be designed to integrate both the knowledge defined in the SWEBOK and the broader AI literacies that are relevant to that knowledge. The paper concludes with some reflections on how this approach might be further developed by educators in the software engineering domain.

II. AI IN THE SWEBOK

Currently, coverage of AI in the SWEBOK appears mostly in the section on "Artificial Intelligence and Machine Learning" (Chapter 16, Computing Foundations, Section 9). Some other relevant material appears in "Testing of and Testing Through Emerging Technologies" (Chapter 5, Software Testing, Section 7), relating to the role of AI, machine learning and deep learning in testing. Coverage of "domain-specific software security" (Chapter 13, "Software Security", section 6.3) includes a brief entry on security for machine learning-based applications, while "computational neurosciences" (Chapter 17, Mathematical Foundations, section 13.1), refers to neural networks, highlighting technological advances that include AI. Finally, the section on "Industry 4.0 and Software Engineering" (Chapter 18, Engineering Foundations, section 10) highlights changes to custom manufacturing and integration with other systems, supported by AI.

This is not to say that AI is not already being applied much more broadly across the SWEBOK knowledge areas, but rather that AI is not explicitly highlighted across every knowledge area to which it might apply. It is not the intention of this article to critique the coverage of artificial intelligence in the SWEBOK nor to consider what other aspects of AI might potentially appear in its coverage. Rather, it aims to take some example knowledge items specifically linked to AI in the SWEBOK and map them to related AI literacies from the SAIL framework. The intent of this mapping is to demonstrate to educators how they could ensure that software engineering students are not only gaining technical competencies in the use of AI but also developing broader literacies that will help them to contextualise and critique their own work from multiple perspectives, ultimately leading to better software engineering practice in the field of AI.

III. THE SCAFFOLDED AI LITERACY (SAIL) FRAMEWORK

The Scaffolded AI Literacy (SAIL) framework has four levels, which are not age- and stage-based but rather provide a pathway of learning. For software engineering students in higher education, we might expect to see outcomes at graduation at levels 3 and 4+ of the framework (Fig. 1).

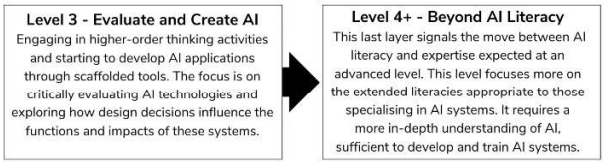

Fig. 1. Levels 3 and 4+ of the SAIL framework

In level 3 - 'evaluate and create AI' - students are engaging in higher-order thinking activities and starting to develop AI applications through scaffolded tools. The focus is on critically evaluating AI technologies and exploring how design decisions influence the functions and impacts of these systems and is the highest level of AI literacy that might be expected of a software engineering generalist. In higher education, this level might usefully be a benchmark that all software engineering students would be expected to have met by the end of their course of study. Level 4+ of the framework is 'beyond AI Literacy', signalling the move from AI literacy to the expertise expected at an advanced level. This level focuses on the highest level of study and requires an in-depth understanding of AI, developing and training AI systems. Level 4+ defines the extended literacies expected from a software engineering graduate with some specialisation in AI.

Aligning these two levels to software engineering graduates is based on the following assumptions:

1. Students completing higher education in software engineering need a fully developed set of AI literacies. Therefore, all software engineering graduates should be able to demonstrate competence at level 3 of the framework, which is presented as the highest level of AI literacy.
2. Students who wish to specialise in AI-related software engineering will need professional competencies appropriate to their level of specialisation. Level 4+, which goes beyond literacy into specialist skill sets, is proposed as being essential for these students.

If levels 3 or 4+ are expected of graduates, levels 1 and 2 (Fig. 2) also have a role to play in the learning journeys of software engineering students.

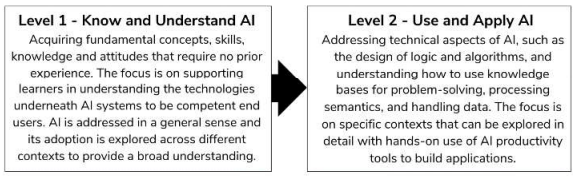

Fig. 2. Levels 1 and 2 of the SAIL framework

Given that the SAIL framework is based on scaffolding learning through all four levels, levels 1 ('know and understand AI') and 2 ('use and apply AI') provide the foundation for the expected graduate levels, so students in their early years of study should be given the opportunity to explore and demonstrate their capabilities at these levels.

A. SAIL framework domains and categories

The framework has six categories of AI literacy, grouped into three domains: "AI concepts", "Application of AI and Technical Skills", and "AI Digital Citizenship". The six categories together ensure an appropriate mix of knowledge, skills, and critical thinking. Fig. 3 shows the domains and categories, with brief descriptions of each.

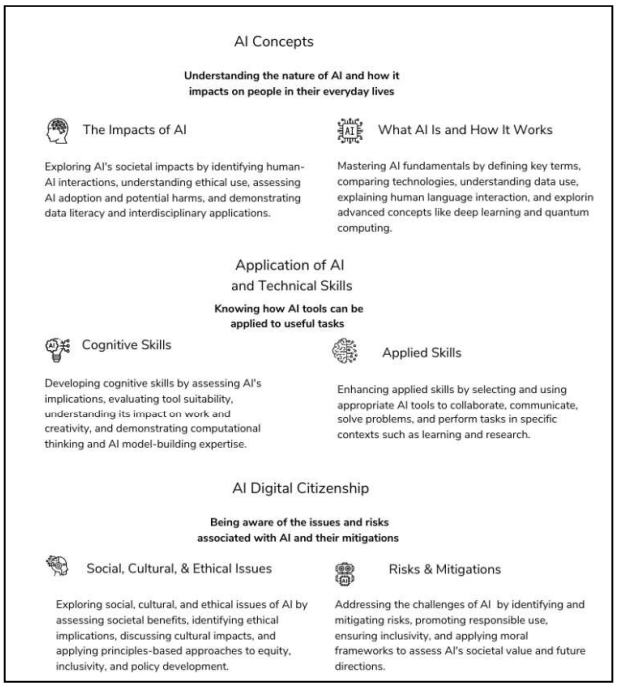

Fig. 3. Domains and categories of the SAIL framework

The following sections of this paper explore some knowledge areas of the SWEBOK that address aspects of AI and align them to relevant competencies from the different levels of the framework, as illustrative examples of how AI literacies can support knowledge areas. These are followed by some example learning activities that would be appropriate for software engineering students to develop a range of competencies in AI literacy and beyond.

IV. Mapping knowledge areas of the SWEBOK to AI literacies

This section illustrates how knowledge areas from the SWEBOK might be mapped to relevant AI literacies. The method for creating these mappings was to first identify all parts of the SWEBOK that explicitly address AI (as briefly discussed in section II). Each of these sections was then examined, and relevant AI literacies from the SAIL framework were identified. Relevant literacies were those that could be explicitly developed in students while they were addressing these knowledge areas. The purpose of the mapping exercise was to demonstrate how the development of key AI literacies can be surfaced to ensure that technical skills are contextualised into a broader context of developing AI literacy.

This article illustrates this process through two examples. The first example shows how the knowledge area of "AI and Software Engineering" can be mapped to AI literacies at levels 3 and 4+ of the framework, while the second example shows how the knowledge area of "Security in ML Applications" can be mapped to levels 1 and 2. These are, of course, just illustrative examples from the broad range of potential mappings that might be considered.

A. Example 1 - AI within the Development Lifecycle

The first example addresses SWEBOK chapter 16, section 9.6, "AI and Software Engineering", which notes that using emerging technologies such as artificial intelligence and machine learning to leverage complex software tools within the development lifecycle adds another dimension to the body of knowledge, whereby AI-based systems may require the development and application of new approaches to the lifecycle.

While the SWEBOK brings an awareness of these issues, exploring related AI literacies can flesh out these concepts into specific competencies. Further, these are not restricted to technical skills but encompass cognitive skills and AI digital citizenship.

As an example, several AI literacy competencies can support an understanding of the changes that AI brings to the software development lifecycle. For level 3 of the framework (which would be expected of all software engineering students regardless of their area of specialisation), in the domain of "Application of AI & Technical Skills", cognitive skills that could apply would include:

- Determine the usefulness of a given AI technology to meet a requirement or perform a task.

Additionally, if model building is involved:

- Understand the steps involved in AI model building, including training, testing, validation, and deployment.

Relevant applied skills could include:

- Evaluate, select, and implement creative approaches to the application of AI across contexts.
- Develop AI projects using appropriate tools scripts, and libraries.

Further, from the domain of "AI Digital Citizenship", the following competency could be considered:

- Evaluate ethical issues related to the design and implementation of AI models and systems (e.g., honesty, intellectual property, and potential harm).

In the category of risks and mitigations, the following also applies:

- Explain strategies for ensuring the accuracy and reliability of AI products.

At level 4+, the components of the framework go beyond AI literacy and apply to those software engineering students who have focus on engineering AI or related systems. For them, the following cognitive skill applies:

- Identify value in successful AI applications, plan and map end-to-end processes from data acquisition to model construction, evaluation, implementation, and life cycle management.

In applied skills, relevant competencies would be:

- Develop AI applications that serve specific purposes in practical settings, leveraging advanced programming and AI techniques.
- Manage projects and collaborate effectively with other groups, leveraging relevant coding and software knowledge to implement ideas.

In the social, cultural, and ethical issues of "AI Digital Citizenship" we should consider:

- Apply ethical considerations to AI projects, such as transparency, explainability, and fairness

and, in risks and mitigations:

- Demonstrate the development and implementation of inclusive AI systems that respect diverse social and cultural contexts.

From this first example, it is possible to see that extending knowledge items from the SWEBOK by matching them with levels of AI literacy increases both the depth and breadth of the topic learning outcomes and contextualises them within the broader concerns of industry and society.

B. Example 2 - Security in Machine Learning Applications

The second example relates to the SWEBOK chapter 13, section 6.3 "Security in Machine Learning-Based Application", which covers AI-specific threats like model poisoning, evasion attacks, and the need for security-aware development. In this example the SWEBOK only provides a brief note, but exploring related AI literacies can build on this to provide opportunities for wider learning and discussion.

The previous example took a detailed section of the SWEBOK and mapped it to levels 3 and 4+ of the framework. In contrast, this example explores how a shorter item from the SWEBOK could be linked to levels 1 and 2 of the framework to provide a shared common understanding about the key issues for all students embarking on their software engineering studies. To expand on the ideas in the SWEBOK, students could investigate how data inputs affect model behaviour, for example by modifying input samples and observing changes in predictions. They might also discuss the real-world impacts of AI security breaches in contexts such as facial recognition, credit scoring, or autonomous vehicles.

For level 1 of the framework - Know and Understand AI - (which is intended to apply to everyone, regardless of their age, stage, and focus of study) the levels of knowledge would be introductory and address the following literacies.

In the domain of "AI Concepts", knowledge of what AI is and how it works that could be developed would include:

- Explain how data is used in different AI systems.

In the domain of "Application of AI & Technical Skills", relevant cognitive skills could include:

- Evaluate the role of data within AI systems and the implications that data has on the training of AI models.

Further, from the domain of "AI Digital Citizenship", the following competencies relating to social, cultural and ethical issues could be considered:

- Identify the ethical implications of AI (e.g., bias, fairness, transparency, accessibility, accountability).

While in the category of risks and mitigations, the following also applies:

- Identify risks presented by AI systems (e.g., security, personal data, privacy, fraud, cyber threats).

At level 2 - Use and Apply AI - AI concepts relating to knowledge of what AI is and how it works relevant to this aspect of the SWEBOK include:

- Apply key terms to explain how AI models are trained and the different steps involved.
- Explain how data is used in AI systems and identify different sources of data used to train various AI models.

In applied skills, a relevant competency would be:

- Explain the different machine learning approaches that can be used (e.g., unsupervised, supervised and reinforcement learning), the role of data in these approaches, and their application to real-world problems.

In "AI Digital Citizenship", social, cultural, and ethical issues, we should consider:

- Understand how bias occurs in AI systems

and, in risks and mitigations:

- Assess the risks associated with data use in AI systems, including issues related to data collection, accuracy, relevance, storage, security, privacy, and potential misuse.

This second example indicates that even a brief reference in the SWEBOK can be expanded out using a set of related AI literacies to engage students in foundational learning activities that can underpin their subsequent advanced learning about AI in software engineering.

V. Example Learning Activities

Mapping knowledge areas to literacies provides an overall vision for how these two components of learning may be related, but on its own is purely theoretical. To put these ideas into practice, it is necessary to embed them into learning activities. Therefore, this section outlines two examples of learning activities that could help students to develop both their understanding of knowledge areas related to AI from the SWEBOK and their AI literacy at various levels.

A. Example Learning Activity 1 - Computing Foundations and Level 3 of SAIL

This learning activity could be suitable for software engineering undergraduates to help them learn how AI is being applied within software engineering, as described in the "Computing Foundations" knowledge area of the SWEBOK addressing AI for SE, related to competencies at level 3 of the SAIL framework ("Evaluate and Create AI").

The activity focuses on how AI can assist in the code review process, which is a key aspect of ensuring software

quality. It involves students simulating the functionality of an AI-supported code review tool, using a simple machine learning platform to highlight the potential of AI in areas like assessing code quality and vulnerability detection.

For this activity, students can be provided with examples of "good" and "bad" code snippets focused on a specific type of vulnerability or style violations. A common, simple example would be the programming style rule that braces should be used with every 'if' statement, regardless of the number of statements within the block [8]. The number of examples should be sufficient to create a classifier. In the activity, students can use a machine learning tool to train a simple model to classify these code examples. Once this model is trained, students can use it to categorise other code examples they create themselves. Depending on the context and the learning goals, students might create their classifiers by coding, for example, by using a Python text classifier library, or for a simpler activity, they could use Google's Teachable Machine image classifier [9] by training the model with images of code samples. While the latter would not be a very realistic tool to use in practice, it offers a quick and simple way to explore the underlying machine learning aspects of creating a code quality classifier. It could also be a first step in an activity to create a more practical classifier using code.

This activity will help students develop competencies within the cognitive skills category at level 3 of the SAIL framework by enabling them to

- Determine the usefulness of a given AI technology to meet a requirement or perform a task.

and, in the applied skills category, to:

- Explain how data sets and training sets are transformed into AI models.
- Develop AI projects using appropriate tools, scripts, and libraries.

Even if using a pre-built tool or a simplified simulation via a platform like Teachable Machine, students will be interacting with, and potentially training (in a simplified manner), an AI model for a specific software engineering purpose.

In the category of risks and mitigations, this activity can help students to:

- Explain strategies for ensuring the accuracy and reliability of AI products.
- Consider the impact of bias in training data and the importance of diverse and inclusive datasets.

By observing the suggestions or feedback provided by their classifiers, students can explore how training data sets may have biases. For example, a style guide model trained on a limited data set might flag some code as non-compliant even if it meets the criteria in the style guide.

More broadly, in the category of what AI is and how it works, students will have the opportunity to:

- Demonstrate an understanding of how AI systems decompose complex problems, how algorithms are developed, and how large datasets are used to train AI models.

By bringing topic learning outcomes such as these, based on AI literacy, into a learning activity, the potential benefits of the learning experience are enhanced.

B. Example Learning Activity 2 – Software Testing and Level 4+ of SAIL

This second learning activity could be suitable for software engineering undergraduates specialising in AI to help them explore how AI can be applied within software testing, as described in the "Software Testing" knowledge area of the SWEBOK, addressing AI in SE, related to competencies at level 4+ of the SAIL framework (Beyond AI Literacy). In this learning activity, students would investigate and prototype the use of AI and machine learning for software testing.

Students could begin by working in groups to select a specific area of software testing where AI techniques can be applied, and then investigate, design, and create a basic prototype or proof-of-concept demonstrating this application. Some possible focus areas could include test case generation (how AI can be used to automatically generate test cases), addressing the test oracle problem (how AI can assist in predicting expected outputs for test cases), evaluation and prioritisation of test cases (how AI can prioritise test cases based on factors like historical data, code changes, or predicted fault likelihood) and bug classification (how AI can analyse bug reports to classify and prioritise them). Students could then review the literature in their chosen area to identify the specific AI techniques that can be used, the benefits and challenges of applying these in software testing, and the data requirements for training and deploying the relevant models.

Having explored the theory, students could then design and implement a basic prototype or proof-of-concept that demonstrates the application of the chosen technique to a simplified software testing scenario. This might involve using existing libraries or platforms, working with a sample dataset, and developing a simplified model or algorithm to perform a relevant testing-related task. They could then evaluate the performance of their prototypes and reflect on the effectiveness of the chosen technique for their specific testing problem, any limitations and potential improvements for their prototype, and the challenges encountered in applying AI for this purpose

This learning activity gives students an opportunity to actively engage in evaluating and creating AI applications in the context of software testing.

Relating to AI literacy, in the category of what AI is and how it works, students will have the opportunity to:

- Demonstrate the ability to learn new AI concepts, tools, and techniques independently, recognising the importance of continual learning in the rapidly evolving field of AI.

It will also help them develop competencies within the cognitive skills category of the SAIL framework by enabling them to:

- Identify value in successful AI applications, plan and map end-to-end processes from data acquisition to model construction, evaluation, implementation, and life cycle management.

and, in the applied skills category, to

- Create AI case studies or projects with measures or hypotheses, recording and analysing the results, and reporting on the findings.

Potentially, depending on how the group work is organised, they might also explore how to:

- Manage projects and collaborate effectively with other groups, leveraging relevant coding and software knowledge to implement ideas.

In the category of risks and mitigations, this activity can help students to begin to:

- Identify and evaluate potential risks associated with AI implementation, including machine-human interaction, intellectual property protection, societal impacts, and misuse of AI.

Although the testing context would not reach deeply into all these aspects, developing an awareness of the potential risks of using AI in the testing context would be an important learning outcome.

It may be noted that both the suggested learning activities discussed above rely to various extents on the creation of data sets. This is a consequence of seeking to provide relatively simple task scenario that provide a straightforward set of options for developing AI literacies. Many alternative learning tasks that develop AI literacies can also be suggested that do not require data sets to be created.

VI. Summary and Conclusions

This article has outlined some ideas for how AI literacy, as expressed in the SAIL framework, can be integrated into software engineering education to enhance graduate skills and awareness in the two broad conceptual aspects of the SWEBOK relating to AI, AI applications in software engineering (AI for SE) and software engineering for AI systems (SE for AI). By mapping knowledge areas from the SWEBOK to AI literacies, the article has suggested how these literacies can provide broader context and understanding to knowledge areas. By providing illustrative examples of learning activities that link knowledge areas to literacies, it suggests how these links can be practically implemented in education.

The full SAIL framework includes 99 AI literacy competencies. The various examples in the article have touched on about one third of these. It is suggested that course delivery in Software Engineering education should take account of the full set of AI literacies to give students the opportunity to develop a comprehensive range of competencies as part of their broader learning about the software engineering body of knowledge. Educators in the software engineering discipline may find it useful to explore the elements of the SAIL framework to identify which AI literacies might be particularly relevant for students in their knowledge domain and professional context.

References

[1] C.K. Sah, L. Xiaoli, M.M. Islam, and M.K. Islam, Navigating the AI Frontier: A Critical Literature Review on Integrating Artificial Intelligence into Software Engineering Education.”” In 2024 36th International Conference on Software Engineering Education and Training (CSEE&T) (pp. 1-5). IEEE.

[2] O. Borges, V. Lenarduzzi, and R. Prikladnicki, “Preliminary insights to enable automation of the software development process in software StartUps,” In Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI CAIN’22: 1st Conference on AI Engineering: Software Engineering for AI (Vol. 10, No. 3522664.3528610). 2022.

[3] E. Nascimento, A. Nguyen-Duc, I. Sundbø, I, and T. Conte, “Software engineering for artificial intelligence and machine learning software: A systematic literature review”. arXiv preprint arXiv:2011.03751. https://arxiv.org/abs/2011.03751. 2020.

[4] H. Washizaki, H. (Ed.), “Guide to the Software Engineering Body of Knowledge v4.0”. IEEE, 2024. https://www.computer.org/education/bodies-of-knowledge/software-engineering/

[5] M. Bublin, S. Schefer-Wenzl, and I. Miladinović, “Educating AI software engineers: Challenges and opportunities. In International Conference on Interactive Collaborative Learning. 2021 (pp. 241-251). Springer International Publishing.

[6] K. MacCallum, D. Parsons, and M. Mohaghegh, (2024). “The Scaffolded AI Literacy (SAIL) Framework for Education: Preparing learners at all levels to engage constructively with Artificial Intelligence”, He Rourou, vol. 1, 23. 2024. https://doi.org/10.54474/herourou.v1i1.10835

[7] K. Stolpe, and J. Hallström, J, “Artificial intelligence literacy for technology education”. Computers and Education Open, vol. 6, 100159. 2024. https://doi.org/10.1016/j.caeo.2024.100159

[8] C. Yang, Y. Liu, and E.J. Yu, “Exploring violations of programming styles: Insights from open source projects”. Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, 2018, p185-189. 10.1145/3297156.3297227

[9] M. Carney, B. Webster, I. Alvarado, K. Phillips, N. Howell, J. Griffith, J. Jongejan, A. Pitaru, and A. Chen, “Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification”. In Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20). ACM, 2020. pp.1–8. https://doi.org/10.1145/3334480.3382839